

Compound Local Binary Pattern para reconhecimento de expressões faciais

André Luiz Dorneles Pires¹, Geovane Menezes Ramos Neto¹

¹Núcleo de Computação Aplicada – Laboratório de Mídias Interativas
Universidade Federal do Maranhão (UFMA) Campus do Bacanga – São Luís – MA

geovane@gmail.com, andre.lp@hotmail.com

Abstract. *Recognition of facial expressions is very important for applications that need to identify the emotional state of the user to obtain the information necessary for the intended application. This study aims to use a technique based on binary local patterns for characterization of the expression sadness when compared to six other expressions. The image base used was the Cohn-Kanade. The best result was 81,90% accuracy for the training/test ratio 60/40 test using as classifier the Support Vector Machine.*

Resumo. *O reconhecimento de expressões faciais é de suma importância para aplicações que necessitam identificar o estado emocional do usuário para obter as informações necessárias a finalidade da aplicação. Este estudo visa utilizar uma técnica baseada em padrões locais binários para caracterização das expressão tristeza quando comparada a outras seis expressões. A base de imagem utilizada foi a Cohn-Kanade. O melhor resultado obtido foi de 81,90% de acurácia para a proporção treino/teste de 60/40 utilizando como classificador a Máquina de Vetores de Suporte.*

1. Introdução

A expressão facial proporciona uma forma não-verbal de comunicação que facilita a reconhecimento de emoções e intenções humanas. A análise automática de expressões faciais tem atraído muita atenção nos últimos anos devido ao seu potencial de aplicação em muitas áreas, tais como interação humano-computador, animação orientada a dados, e aplicações personalizadas para produtos de consumo [Jabid et al. 2010]. Embora muito trabalho tem sido feito, propôr uma abordagem eficiente e discriminante de representação de características faciais que possa ser robusta as variações na iluminação, pose, e outras mudanças ainda é uma tarefa desafiadora [Shan et al. 2009]. Comumente utiliza-se técnicas de extração de características faciais incluindo métodos geométricos baseados na aparência [Tian et al. 2005].

Recentemente, a análise de expressões faciais baseada em *Local Binary Pattern* (LBP) [Shan et al. 2009], [Shan et al. 2005] e suas variantes ganharam muita popularidade por seus desempenhos superiores. O operador LBP foi originalmente introduzido para análise de textura [Shan et al. 2005] e depois este método tem sido aplicado com êxito na autenticação de face e reconhecimento de expressões faciais. O método LBP extrai informações de textura local. O LBP é utilizado com a comparação entre as intensidades dos níveis de cinza dos P pixels vizinhos com a intensidade do pixel central sendo este eficiente e robusto para a variação de iluminação monotônica.

Embora o LBP forneça uma teoria simples e uma abordagem eficiente para a análise de expressões faciais, este possui algumas limitações. Em primeiro lugar, ele mostra um baixo desempenho na presença de ruído aleatório [Zhou et al. 2008]. Para abordar esta questão, *Local Ternary Pattern* (LTP) [Tan and Triggs 2010] tem sido apresentado com um nível de discriminação adicional quando comparado ao LBP, este aumenta a robustez contra o ruído em regiões uniformes e não uniformes. Em segundo lugar, o método LBP considera apenas o sinal da diferença entre dois valores de intensidade de nível de cinza e descarta assim a magnitude da diferença, que é uma informação de textura muito importante.

Para explorar a informação de magnitude, Jabeed et al [Jabid et al. 2010] introduziu *Local Directional Pattern* (LDP). Em vez de valores de cinza, o LDP emprega a magnitude dos valores da borda em diferentes direções a fim de codificar as informações de uma região local. No entanto, LDP ainda gera inconsistentes códigos de uniforme em regiões uniformes e lisas e depende fortemente do número de direções de borda proeminentes.

Neste trabalho, utilizamos o *Compound Local Binary Pattern* (CLBP), que é um método de representação de características baseado no operador LBP original, para o reconhecimento de expressões faciais. Ao contrário do operador LBP original que utiliza P bits para codificar apenas os sinais das diferenças entre o pixel central e os P valores de cinza dos pixels vizinhos, o método proposto emprega 2P bits, em que os P bits adicionais são usados para codificar as informações da magnitudes das diferenças entre o centro e os valores de níveis de cinza vizinhos em uma vizinhança local utilizando uma limiar. A motivação por trás do esquema de codificação proposto é aumentar a robustez da representação de características por incorporar informações locais adicionais que são descartadas pelo operador LBP original.

O desempenho da representação característica do CLBP é avaliado em termos de taxa de classificação usando a máquina de vetores de suporte (MVS). Os experimentos foram realizados utilizando o banco de dados de expressão facial Cohn-Kanade (CK) [Kanade et al. 2000]. Os experimentos demonstram que, o operador CLBP é mais robusto para extrair e caracterizar informações facial e proporciona maior taxa de classificação em comparação com algumas técnicas de representação de característica existentes.

2. Referencial Teórico

Este capítulo apresenta os conceitos básicos para o embasamento teórico da pesquisa aqui exposta. São abordados os conceitos de processamento digital de imagens, descritores de textura, exemplos de operadores que descrevem textura, sendo eles o *Local Binary Pattern* (LBP) e a variantes deste, sendo esta o *Compound Local Binary Pattern* (CLBP) e também trataremos sobre o classificador Máquina de Vetores de Suporte (MVS).

2.1. O processamento digital de imagens

O processamento digital de imagens são processos cujas entradas e saídas são imagens e, além disso, engloba processos que extraem atributos de imagens bem como o reconhecimento de objetos individuais. O processamento digital de imagens pode ser dividido em processos computadorizados. Processos de baixo, médio e alto nível, esses processos são diferenciados de acordo com o seu objetivo [Gonzalez and Woods 2006].

Processos de baixo nível envolvem operações primitivas como pré-processamento de imagem para reduzir ruídos, melhorar contrastes e tornar a imagem mais nítida. Um processo de baixo nível é caracterizado pelo fato que sua entrada e saída são imagens. Processos de médio nível em imagens envolvem tarefas como segmentação (particionar uma imagem em regiões ou objetos), descrições desses objetos para reduzi-los para uma forma conveniente para um processamento computadorizado, e classificação (reconhecimento) de objetos individuais. Um processo de nível médio é caracterizado pelo fato que suas entradas geralmente são imagens, mas suas saídas são atributos extraídos dessas entradas (por exemplo, arestas, contornos, ou uma característica que diferencia um objeto do outro). Finalmente, processamentos de alto nível envolvem extrair características dos objetos que os façam ser melhor compreendidos ou desempenhando as funções cognitivas normalmente associadas a visão.

2.2. Descritores de Textura

A textura pode ser utilizada na análise de imagens de várias maneiras: na segmentação de objetos distintos e regiões em cenários, na classificação ou reconhecimento na superfície de materiais ou no cálculo da forma da superfície [Bovik et al. 1990].

Descritores de textura tem o objetivo de caracterizar a textura das imagens buscando representar texturas semelhantes de formas semelhantes utilizando propriedades da textura, extraindo informações a partir de análises estatísticas, geométricas, estruturais, baseadas em modelo ou processamento de sinais [Tuceryan et al. 1993].

Um grande problema na análise da textura consiste na complexidade de texturas em aplicações práticas, onde normalmente estão sujeitas a mudanças de orientação, escala e/ou aparências visuais. Além disto, o grau de complexidade computacional de vários descritores de textura é bem alto [Ojala et al. 1996] conforme o estudo comparativo de Randen e Husoy [Randen and Husoy 1999].

2.3. Local Binary Pattern (LBP)

A técnica LBP (*Local Binary Pattern*) [Ojala et al. 1996] apresentada por Timo Ojala, neste trabalho ele propõe uma abordagem teórica e computacionalmente simples, que é robusta em termos de variações nos níveis de cinza e que discrimina efetivamente uma larga gama de texturas rotacionadas. O LBP é um operador de textura invariante aos níveis de cinza e rotação, baseado em padrões locais binários.

Inicialmente devemos definir um arranjo de vizinhanças circularmente simétricas de P membros em um círculo de raio R . Dado este arranjo de vizinhanças, podemos definir a textura T em uma vizinhança local de uma imagem de textura monocromática como a distribuição conjunta dos níveis de cinza de P ($P \geq 1$) pixels da imagem.

$$T = t(g_c, g_0, \dots, g_{P-1}) \quad (1)$$

onde g_c é o nível de cinza do pixel central e $g_P(g_0, \dots, g_{P-1})$ corresponde aos valores de P igualmente espaçados em um círculo de raio R ($R \geq 0$). Se as coordenadas de g_c são $(0, 0)$ então as coordenadas de g_P são dadas por $(-R \sin \frac{2\pi p}{P}, R \cos \frac{2\pi p}{P})$ (Figura 1).

O LBP é então calculado como:

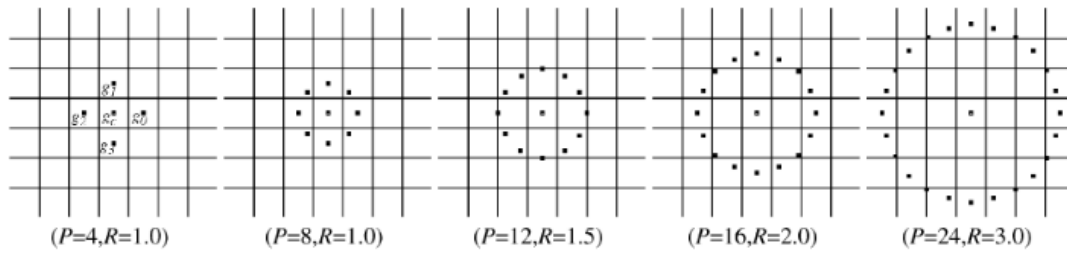


Figura 1. Vizinhanças circularmente simétricas para diferentes valores de P e R . Fonte: Retirado de [Ojala et al. 2002].

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2)$$

onde $s(x)$ é uma função que devolve 1 se $x \geq 0$ ou 0, caso contrário.

Um exemplo do cálculo do LBP é apresentado na Figura 2.

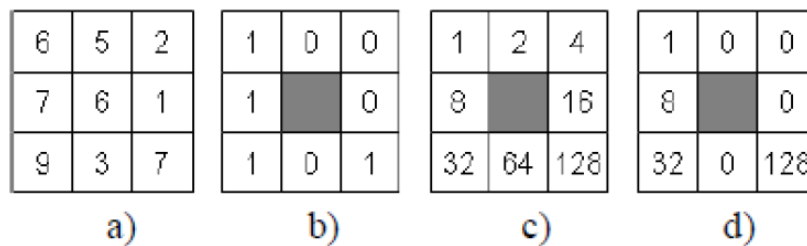


Figura 2. Cálculo do LBP. (a) A imagem; (b) A imagem binária; (c) Matriz de Pesos; (d) Valores resultantes. Fonte: Retirado de [da Rocha 2014].

Dada uma janela 3x3 (Figura 2a), os valores do pixel central é subtraído de cada um dos valores dos *pixels* vizinhos, o resultado da subtração é utilizado na função $s(x)$ que substituirá o valor anterior do *pixel* vizinho por 0 ou 1, dependendo do retorno da função $s(x)$ (Figura 2b). Cada valor da matriz binária é multiplicado pela sua respectiva posição na matriz de pesos (Figura 2c). O LBP é o resultado da soma de todos os valores resultantes das multiplicações (Figura 2d). O exemplo em questão tem como resultado do LBP o valor de 169.

2.4. Compound Local Binary Pattern (CLBP)

Apesar do LBP ser teoricamente simples e ainda assim ser eficiente, esta técnica utiliza somente a informação gerada pelos sinais das diferenças entre o pixel central e os vizinhos e descarta a magnitude dessa diferença, que é uma importante informação sobre a textura.

O *Compound Local Binary Pattern* (CLBP) [Ahmed et al. 2011] é uma extensão do operador LBP. Ao contrário do LBP, que usa P bits para codificar somente os sinais das diferenças entre o nível de cinza do pixel central e os níveis de cinza de P vizinhos, o CLBP emprega $2P$ bits, onde os P bits adicionais são utilizados para codificar a magnitude da diferença entre os níveis de cinza já citados.

A motivação do CLBP é aumentar a robustez da representação de características incorporando informação local adicional que é descartada pelo LBP tradicional. Além de utilizar os P bits do LBP, o CLBP gera mais P bits de acordo com a Equação 3.

$$s(i_p, i_c) = \begin{cases} 1 & \text{se } |i_p - i_c| > M_{avg} \\ 0 & \text{caso contrário.} \end{cases} \quad (3)$$

onde M_{avg} é média da magnitude da diferença entre i_p e i_c na vizinhança local.

Em uma vizinhança 3×3 , o CLBP opera com 8 vizinhos ao redor do *pixel* central, gerando um código de 16 bits. Para reduzir o número de características, esse código é dividido em dois códigos sub-CLBP. Cada sub-CLBP é obtido concatenando os valores da sequência de bits (1, 2, 5, 6, ..., $2P - 3$, $2P - 2$) e (3, 4, 7, 8, ..., $2P - 1$, $2P$), respectivamente do código $2P$ bits original do CLBP, representado na Figura 3.

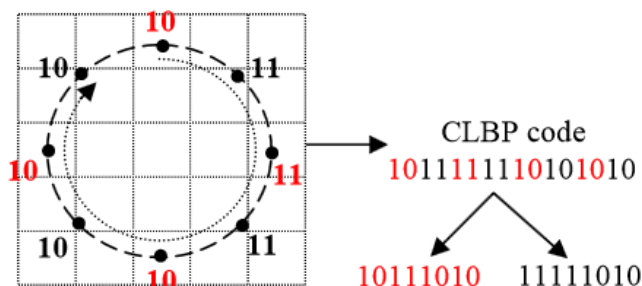


Figura 3. Geração dos códigos sub-CLBP a partir do código original do CLBP. Retirado de [Ahmed et al. 2011].

Os dois códigos sub-CLBP são tratados como códigos binários separados, sendo estes calculados separadamente. A partir das imagens geradas, extraímos os histogramas e concatena-se ambos gerando o vetor de características.

2.5. Máquina de Vetores de Suporte

A máquina de vetores de suporte (MVS) é um método de aprendizagem supervisionado usado para estimar uma função que classifique dados de entrada em duas classes, sendo que o princípio básico por trás da MVS é a construção de um hiperplano que sirva como superfície de decisão, em que a margem de separação entre as classes seja máxima. Assim, a finalidade do treinamento através da MVS é a obtenção de hiperplanos que dividam as amostras de tal modo que sejam otimizados os limites de generalização [Vapnik 2013].

Um hiperplano pode ser compreendido como uma superfície de separação de duas regiões em um espaço multidimensional, em que o número de dimensões possíveis pode ser muito grande ou até mesmo infinito. Mesmo quando as duas classes não são separáveis, a MVS é capaz de encontrar um hiperplano através do uso de conceitos pertencentes a teoria da otimização. Porém, quando o conjunto de amostras é composto por duas classes separáveis, um classificador MVS é capaz de encontrar um hiperplano baseado em um conjunto de pontos, chamados vetores de suporte, o qual maximiza a margem de separação entre as classes [Vapnik 2013].

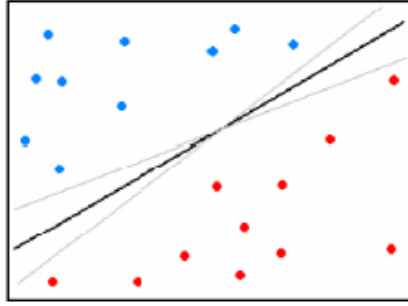


Figura 4. Separação de duas classes através de duas classes de hiperplanos.
Fonte: Retirado de [da Rocha 2014].

A Figura 4 exemplifica hiperplanos de separação entre duas classes linearmente separáveis. A linha central representa o hiperplano ótimo, pois separa as duas classes e matem a maior distância possível com relação aos pontos da amostra.

Dado o conjunto de treinamento (x_i, y_i) , sendo $x_i \in \mathbb{R}^n$ o vetor de entrada, y_i a classificação correta das amostras e $i = 1, \dots, n$ o índice de cada ponto amostral. O objetivo da classificação é estimar a função $f: \mathbb{R}^n \rightarrow \{\pm 1\}$, que separe corretamente os exemplos de teste em classes distintas [Vapnik 2013].

A etapa de treinamento estima a função $f(x) = (w \cdot x) + b$, procurando por valores de w e b tais que a relação, representada pela Equação 4 seja satisfeita.

$$y_i((w \cdot x_i) + b) \geq 1 \quad (4)$$

onde w é o vetor normal ao hiperplano de decisão e b o corte ou distancia da função f em relação a origem. Os valores ótimos de w e b serão encontrados ao minimizar a Equação 5, de acordo com a restrição dada pela Equação 4 [Chaves 2006].

$$\Phi(w) = \frac{w^2}{2} \quad (5)$$

A MVS ainda permite encontrar um hiperplano que minimize a ocorrência de erros de classificação incluindo variáveis de folga, as quais permitem que as restrições presentes na equação 4 sejam quebradas.

Dessa forma o problema passa a ser de minimização da Equação 6, conforme a restrição imposta pela Equação 4, no qual C é um parâmetro de treinamento que irá determinar um equilíbrio entre a complexidade do modelo e o erro de treinamento, sendo este escolhido pelo usuário [Vapnik 2013].

$$\Phi(w, \xi) = \frac{w^2}{2} + C \sum_{i=1}^N \xi_i \quad (6)$$

$$y_i((w \cdot x_i) + b) + \xi_i \geq 1 \quad (7)$$

A Equação 8 é alcançada através da teoria dos multiplicadores de Lagrange, cujo

objetivo é encontrar os multiplicadores de Lagrange α_i ótimos que satisfaçam a Equação 9.

$$w(\alpha) = \sum_{i=1}^N \alpha_i - 1/2 \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (x_i, x_j) \quad (8)$$

$$\sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C \quad (9)$$

Somente para os pontos onde a restrição da Equação 4 seja exatamente igual unidade terá correspondente $\alpha \neq 0$. Estes pontos são chamados de vetores de suporte, pois se localizam geometricamente sobre as margens.

Os pontos que representam os vetores de suporte estão destacados na Figura 5. Os pontos além da margem não influenciam decisivamente na determinação do hiperplano, enquanto os vetores de suporte, por terem pesos não nulos, são decisivos [Vapnik 2013].

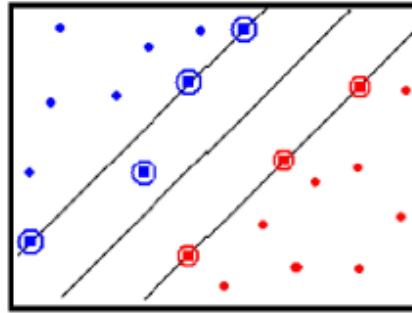


Figura 5. Vetores de Suporte para determinação do hiperplano de separação (destacados por círculos). Fonte: Retirado de [da Rocha 2014].

Há ainda os casos em que as amostras não são linearmente separáveis, a MVS necessita de uma transformação não-linear que transforme o espaço de entrada (dados) para um novo espaço (espaço de característica). Este espaço deve ter dimensão suficientemente grande e por meio deste a amostra possa ser linearmente separável. Esta transformação depende do cálculo de uma função K de núcleo de um produto interno.

A Equação 10 apresenta o resultado da Equação 8 com a utilização de um núcleo K .

$$w(\alpha) = \sum_{i=1}^N \alpha_i - 1/2 \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (10)$$

Há uma grande variedade de funções de núcleo, podendo encontrada-las em uma variedade de aplicações. Nesta monografia a *Radial Basis Function* (RBF) $\exp(-\gamma \|x_i - x_j\|^2)$ será usada.

3. Metodologia

Este capítulo descreve os procedimentos utilizados neste trabalho para metodologia da caracterização da expressão tristeza quando comparada a outras seis expressões faciais (Alegria, Aversão, Medo, Raiva, Neutro, Surpresa) utilizando o *Compound Local Binary Patterns*. A Figura 6 apresenta as etapas do estudo, que são: Aquisição de Imagens, Decomposição da ROI, Extração de Características e Reconhecimento de Padrões.

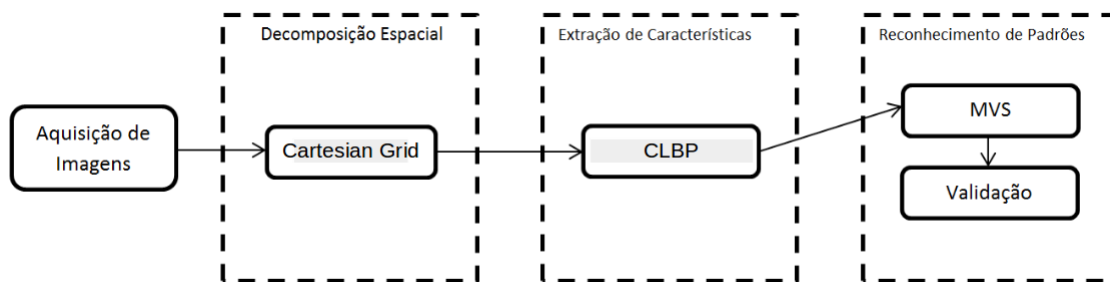


Figura 6. Representação da metodologia deste estudo

3.1. Aquisição de Imagens

O desempenho do método proposto é avaliado com uma base de imagens bem conhecida, a Base de Dados de Expressão Facial Cohn-Kanade [Kanade et al. 2000]. A base de imagens contém 1224 imagens de faces de um conjunto de seis expressões faciais (Alegria, Aversão, Medo, Raiva, Tristeza e Surpresa), utilizamos além destas um sétimo tipo de expressão denominado Neutro, também contido na base.

Neste trabalho utilizamos um subconjunto de 876 Regiões de Interesse, sendo 126 expressões de tristeza e 750 das outras seis expressões. A Figura 7 apresenta exemplos de ROIs utilizadas neste trabalho.



Figura 7. (a) representa uma expressão facial tristeza (b) e (c) representam as expressões de aversão e medo respectivamente.

3.2. Decomposição Espacial

O objetivo desta etapa é gerar uma representação mais adequada para o propósito da análise de textura. Nela, a ROI é o resultado da aplicação de uma abordagem de divisão espacial *Cartesian Grid* que será usada como entrada para o CLBP.

A *Cartesian Grid* visa dividir a ROI como uma grade. A função é parametrizada por m e n , onde m é o número de divisões horizontais e n é o número de divisões verticais.

A quantidade de ROIs resultantes é dado por $m \times n$. Neste trabalho, utilizamos a *Cartesian Grid* 3x3.

A Figura 8a representa a abordagem *Cartesian Grid* 2x2, a Figura 8b representa a ROI original com a *grid* 4x4. A Figura 8c e d são exemplos de quadrantes obtidos do *Cartesian Grid*.

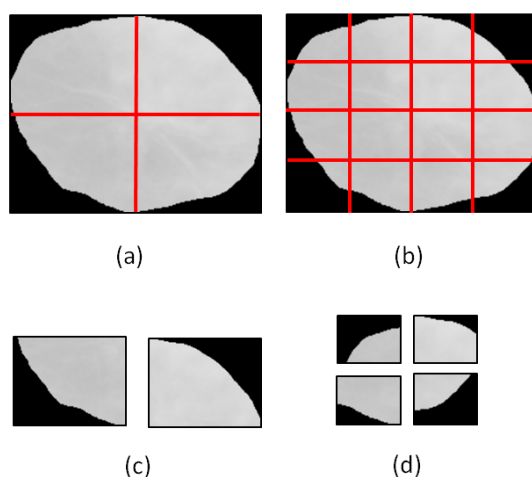


Figura 8. Exemplos de *Cartesian Grid*. Em (a) 2x2, em (b) 4x4 e em (c) e (d) exemplos das ROIs resultantes.

3.3. Extração de Características

A técnica *Compound Local Binary Pattern* (CLBP), foram aplicadas para todas as imagens resultantes da aplicação da decomposição espacial.

A técnica CLBP foi calculada em janela 3x3 e o *threshold* τ_1 utilizado no cálculo do CLBP foi utilizado como a média dos módulos das diferenças entre o centro e os valores de nível de cinza vizinhos.

3.4. Reconhecimento de Padrões

Com o objetivo de fazer a validação dos resultados obtidos medindo o desempenho de cada um dos pares abordagem/técnica com o auxílio de algumas estatísticas, esta subseção foi incluída neste estudo e é detalhada em seguida.

Conforme descrito na Sessão 2.5, o objetivo desta etapa consiste em classificar cada expressão em tristeza ou não tristeza.

Os experimentos foram realizados de acordo com o fluxo de atividades descrito pela Figura 9. A base de características foi normalizada entre -1 e 1. Esse processo tem como objetivo padronizar a distribuição de valores das variáveis, que podem assumir diferentes domínios. Além disso, busca ajudar o classificador a convergir com maior facilidade na etapa de treinamento. Portanto, utilizou-se o *svm-scale* presente no pacote *libSVM*¹ [Chang and Lin 2011] para realizar esta tarefa.

¹Este pacote contém a implementação da MVS utilizada neste trabalho

Após a etapa de normalização, a base de característica foi dividida em base de treino e base de teste. Este trabalho adotou vários critérios para a divisão entre base de treino e teste, a saber: 50/50, 60/40, 70/30 e 80/20.

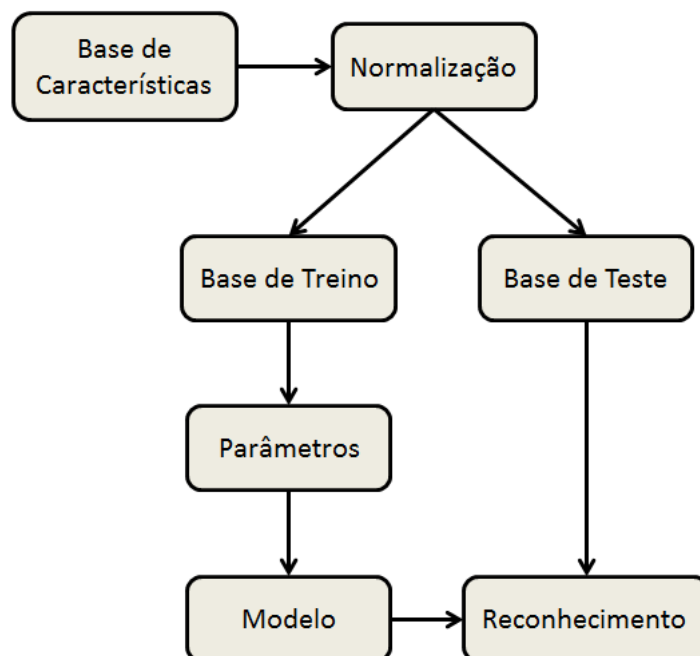


Figura 9. Fluxo da metodologia para reconhecimento de padrões.

A divisão treino e teste foi realizada 3 vezes aleatoriamente para eximir o processo de reconhecimento de efeitos de correlação. Cada base de treinamento teve parâmetros estimados. Foram estimados os parâmetro de custo C e o grau de complexidade da função de mapeamento γ utilizados pela MVS. Para tanto, usou-se o *script* em *python* conhecido como: *grid.py* presente no pacote libSVM. Este *script* busca através da validação cruzada, a melhor combinação de parâmetros para a base que retorne como resposta o melhor percentual de acerto total sobre as amostras de treino e teste [da Rocha 2014].

Ao final da etapa de treinamento é produzido o modelo (função) com os vetores de suporte que a MVS utilizará para classificar as amostras de teste. Desta forma, esta construção de classificação busca se aproximar as condições reais de teste. Então o modelo gerado por ser utilizado para realizar a etapa de reconhecimento de padrões com amostras de teste separadas.

4. Resultados

Para os testes foi utilizado a base Cohn-Kanade [19]. As ROIs foram extraídas usando o *Haar Cascade*. Os testes foram realizados usando SVM. A SVM foi configurado com núcleo radial e teve o parâmetro de custo ajustado para 8.0 e γ para 2^{-8} .

O resultado obtido foi de 81,90% de Acurácia. Os resultados adquiridos se mostram promissores, pois mesmo sem uma escolha otimizada dos parâmetros do CLBP a metodologia obteve bons resultados quando compara a trabalhos já existentes. Os resultados indicam uma boa discriminação entre expressões de tristeza quando comparada com

as outras seis expressões, indicando este trabalho propõe uma boa metodologia e que só tende a aumentar as chances de um reconhecimento bem sucedido.

5. Conclusão e Próximos Passos

Este trabalho apresentou uma metodologia para descrição dos padrões de expressões faciais usando como extrator de características o CLBP e a técnica de divisão espacial *Cartesian Grid 3x3*.

A utilização da *Cartesian Grid* forneceu certo grau de localização espacial quando utilizada com o CLBP. Se utilizarmos somente o CLBP sem a utilização de um divisor espacial os resultados teriam uma queda de desempenho graças falta de informações espaciais válidas.

Referências

- Ahmed, F., Hossain, E., Bari, A., and Shihavuddin, A. (2011). Compound local binary pattern (clbp) for robust facial expression recognition. In *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*, pages 391–395, Obuda University, Budapest, Hungary. IEEE.
- Bovik, A. C., Clark, M., and Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1):55–73.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.
- Chaves, A. d. C. F. (2006). *Extração de Regras Fuzzy para Máquinas de Vetores Suporte (SVM) para Classificação em Múltiplas Classes*. PhD thesis, PUC-Rio.
- da Rocha, S. V. (2014). *Diferenciação do Padrão de Malignidade e Benignidade de Massas em Imagens de Mamografias Usando Padrões Locais Binários, Geoestatística e Índices de Diversidade*. PhD thesis, Universidade Federal do Maranhão.
- Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, NJ, USA.
- Jabid, T., Kabir, M. H., and Chae, O. (2010). Robust facial expression recognition based on local directional pattern. *ETRI journal*, 32(5):784–794.
- Kanade, T., Cohn, J. F., and Tian, Y. (2000). Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 46–53. IEEE.
- Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59.
- Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987.
- Randen, T. and Husoy, J. H. (1999). Filtering for texture classification: A comparative study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(4):291–310.

- Shan, C., Gong, S., and McOwan, P. W. (2005). Robust facial expression recognition using local binary patterns. In *Image Processing, 2005. ICIIP 2005. IEEE International Conference on*, volume 2, pages II–370. IEEE.
- Shan, C., Gong, S., and McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816.
- Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing, IEEE Transactions on*, 19(6):1635–1650.
- Tian, Y.-L., Kanade, T., and Cohn, J. F. (2005). Facial expression analysis. In *Handbook of face recognition*, pages 247–275. Springer.
- Tuceryan, M., Jain, A. K., et al. (1993). Texture analysis. *Handbook of pattern recognition and computer vision*, 2:207–248.
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media, New York, NY, USA.
- Zhou, H., Wang, R., and Wang, C. (2008). A novel extended local-binary-pattern operator for texture analysis. *Information Sciences*, 178(22):4314–4325.